AFRL-AFOSR-VA-TR-2016-0368

A COMPREHENSIVE TOOLSET FOR GENERAL-PURPOSE PRIVATE COMPUTING AND OUTSOURCING

Marina Blanton
UNIVERSITY OF NOTRE DAME DU LAC

12/08/2016
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTA2
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
13-12-2016	Final Performance	01 Mar 2013 to 31 Aug 2016

4. TITLE AND SUBTITLE
A COMPREHENSIVE TOOLSET FOR GENERAL-PURPOSE PRIVATE COMPUTING AND OUTSOURCING

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-13-1-0066

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Marina Blanton

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
UNIVERSITY OF NOTRE DAME DU LAC
940 GRACE HALL
NOTRE DAME, IN 465565602 US

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AF Office of Scientific Research
875 N. Randolph St. Room 3112
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR RTA2

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-AFOSR-VA-TR-2016-0368

12. DISTRIBUTION/AVAILABILITY STATEMENT
A DISTRIBUTION UNLIMITED: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The over-reaching goal of this project is to provide the necessary tools and techniques for supporting general-purpose secure computation and outsourcing. The three main thrusts of the project are: (i) development of efficient techniques for securely working with standard data types, (ii) designing efficient data-oblivious algorithms and data structures suitable for secure computation and outsourcing, and (iii) building a compiler for translating a program written in a conventional programming language which is intended to handle private data into the corresponding secure distributed implementation that provably protects private data throughout program execution. This report summarizes the research findings of the project and scientific advances made towards each of the research thrusts throughout the project duration.

15. SUBJECT TERMS
SECURE OUTSOURCED COMPUTATION

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF

a. REPORT	b. ABSTRACT	c. THIS PAGE		PAGES	19a. NAME OF RESPONSIBLE PERSON
Unclassified	Unclassified	Unclassified	UU		NGUYEN, TRISTAN
					19b. TELEPHONE NUMBER *(Include area code)* 703-696-7796

A Comprehensive Toolset for General-Purpose Private Computing and Outsourcing

Marina Blanton
Computer Science and Engineering Department
University at Buffalo

Abstract

The over-reaching goal of this project is to provide the necessary tools and techniques for supporting general-purpose secure computation and outsourcing. The three main thrusts of the project are: (i) development of efficient techniques for securely working with standard data types, (ii) designing efficient data-oblivious algorithms and data structures suitable for secure computation and outsourcing, and (iii) building a compiler for translating a program written in a conventional programming language which is intended to handle private data into the corresponding secure distributed implementation that provably protects private data throughout program execution. This report summarizes the research findings of the project and scientific advances made towards each of the research thrusts throughout the project duration.

1 Project Objectives

Cloud computing enables convenient on-demand access to computing and data storage resources, which can be configured to meet unique clients' constraints and utilized with minimal management overhead. The recent rapid growth in availability of cloud services makes them attractive and economically sensible for clients with limited computing or storage resources who are unable to procure and maintain their own computing infrastructure. This includes numerous applications in commercial, government, and military domains, including, e.g., weak devices such as sensors operating outside the base. One of the largest possibilities that the cloud enables is computation outsourcing, when the client can utilize any necessary computing resources for its computational task. Security considerations, however, stand on the way of harnessing the full benefits of cloud computing to the fullest extent and prevent clients from placing their sensitive data or computations on the cloud. This is of utmost importance for data concerning national security, but even in non-military contexts businesses are also hesitant to make their proprietary available to the cloud [1]. While in general sensitive data can be protected by the means of encryption, traditional encryption is not suitable for computation over data. Protection of the data in outsourced computation was thus set to be one of the main goals of this research.

The broad goal of this research project is to develop techniques suitable for secure and general data processing and outsourcing. The desire to carry out computation in a privacy-preserving manner without revealing information about the sensitive inputs throughout the computation is not new: it has been a topic of research since Yao's seminal work on secure function evaluation [9]. However, despite the cheer volume of research literature on privacy-preserving computation and newly appearing secure outsourcing techniques, most of the previously available techniques focused on rather narrow domains such as integer-based arithmetic, keyword search over encrypted data, or

two-party set operations. Little or no attention has been paid to other types of computation, as well as to data structures and algorithms suitable for secure data processing in untrusted environments.

Another reason why secure computation techniques are not commonly used in practice is their complexity and overhead. Recent progress in the performance of secure multi-party computation techniques, however, demonstrated that secure computation can be very fast (e.g., millions of operations per second performed by Sharemind on a LAN). This, combined with the shift toward cloud computing and storage, offered a major incentive for further development of new techniques for *general-purpose* secure data processing. We therefore believed that it was the prime time to enable privacy-preserving execution of any functionality or program, and the grand goal of this research was to *develop techniques for securely computing on data of different types and their collections, including oblivious data structures and algorithms.* Data-oblivious execution is defined as having the sequence of executed instructions and the sequence of accessed memory locations to be independent of the input.

Toward that goal, this research intended to cover new techniques for major data types and collections, such as boolean, integer, and real values, strings, sets, vectors, and matrices. Furthermore, to facilitate the use of secure general-purpose computing, research was needed to develop data-oblivious algorithms and data structures for common tasks such as search and graph algorithms. Note that the great majority of data structures and algorithms commonly used in practice are not data oblivious, while naive approaches for achieving data-obliviousness incur a substantial increase in computation time over best-known solutions (compare, for instance, non-oblivious logarithmic-time binary search with oblivious linear-time scan).

We make a distinction between the party or parties who hold private inputs and computational parties who conduct the computation. This allows the framework to be used in many contexts including secure joint computation by multiple parties and computation outsourcing by one or more parties. Our techniques are information-theoretically secure and promise to be particularly efficient and suitable for large-scale applications.

To foster adoption of our and previously developed techniques, another goal of this project was to build a compiler that translates a program written in a high-level C-like language to an executable which can be securely evaluated by a number of parties. The goal was to support as wide of a range of functionalities as possible, i.e., as long as the functionality known at the run-time, it can be securely evaluated in our framework. Performance of compiled programs was intended to be evaluated on a number of diverse applications including statistical analysis and biometric processing, as well as commonly used operations and data structures, which is of high relevance to the government, military, and commercial sectors.

2 Project Research Results

In this section we summarize research findings of the project. The description is structured according to the three main thrusts of the project, which are: (i) development of efficient techniques for securely working with standard data types, (ii) designing efficient data-oblivious algorithms and data structures suitable for secure computation and outsourcing, and (iii) building a compiler for translating a program written in a conventional programming language which is intended to handle private data into the corresponding secure distributed implementation that provably protects private data throughout program execution.

Research publications associated with this project are [4, 7, 13, 10, 11, 6, 5, 3]. All of them accomplish support from this research grant.

2.1 Support for Secure Processing of Standard Data Types

Previously, research on securely handling private data almost entirely focused on integer operations. Secure and efficient implementations of integer arithmetic could also be used to implement Boolean operations and string manipulations with strings represented as arrays of integer values. Support for proper floating-point operations on private real numbers was, however, lacking and closing this gap has been the focus of this research. As part of this research, we

1. designed efficient secure multi-party techniques for floating-point computation in a standard linear secret sharing framework. This includes a variety of operations such as addition, subtraction, multiplication, division, comparisons, rounding, conversion to and from integers, and supplemental operations.

2. designed efficient (and fast converging) secure protocols for complex operations over real numbers such as square root, logarithm, and exponentiation.

3. evaluated the developed and existing techniques for integer, fixed point, and floating point arithmetic and demonstrated efficiency of the developed protocols despite complexity of the operations.

Details of the design and implementation are available from [4]. Consequently, we applied this design to secure two-party computation techniques based on homomorphic encryption for the setting where alternative frameworks are not an option [2]. In our further research, we strengthened the security guarantees of the constructions to be resilient to adversarial behavior in the strongest security model through a number of novel protocols and zero-knowledge techniques [3].

We also extended our work on private and data-oblivious set and multiset operations [5] and published new techniques on secure and verifiable matrix multiplication outsourcing [10].

2.2 Data-Oblivious Algorithms

Our work on data-oblivious algorithms primarily focused on graph algorithms. Graph algorithms are fundamental in computer science and are used in a variety of applications. Given a graph $G = (V, E)$ as the input, our solutions use an adjacency matrix representation of the graph. This representation has size $\Theta(|V|^2)$ and is asymptotically optimal for dense graphs with $|E| = \Theta(|V|^2)$. We developed a number of novel data-oblivious graph algorithms for classical graph problems which lead to secure constructions for evaluating such problems in secure multi-party computation or secure outsourcing settings. In particular, we designed the following data-oblivious algorithms:

- breadth-first search (BFS) of complexity $O(|V|^2)$,
- single-source single-destination (SSSD) shortest path of complexity $O(|V|^2)$,
- minimum spanning tree of complexity $O(|V|^2)$,
- maximum flow of complexity $O(|V|^3|E|\log(|V|))$,
- and maximum matching size in bipartite graphs of complexity $O(|V|^3\log(|V|))$.

The details of our techniques are available from [7, 6]. Our research also treated data-oblivious data structures as described in the next section.

2.3 Compiler for Secure Distributed Computation

As part of the third thrust, we introduced PICCO (Private dIstributed Computation COmpiler) — a system for translating a general-purpose program for computing with private data into its secure implementation and executing the program in a distributed environment. The main component of PICCO is a source-to-source compiler that translates a program written in an extension of the C programming language with provisions for annotating private data to its secure distributed implementation in C. The C language was chosen due to its popularity and, more importantly, performance reasons. The resulting program can consequently be compiled by the native compiler and securely run by a number of computational nodes in the cloud or similar environment. Besides the compiler, PICCO includes programs that aid secure execution of user programs in a distributed environment by preprocessing private inputs and recovering outputs at the end of the computation.

The techniques underlying PICCO's secure execution build on a threshold linear secret sharing scheme for representation of and secure computation over private values. This setting was chosen to due its flexibility (i.e., permitting both secure multi-party computation and secure computation outsourcing) and speed. Thus, secure execution of the compiled programs is performed by $n > 2$ computational parties.

The compiler supports all features of the C language and all programs that do not result in revealing information about private values. For example, the number of loop iterations in a program cannot depend on private values as this information cannot be revealed even at program runtime, data flow from a private to a public variable is not permitted, and functions executed within a conditional statement with a private condition are not permitted to have public side effects. The original PICCO design in [13] did not support the use of C pointers (and thus features such as dynamic memory allocation), and this limitation has been consequently mitigated in [11].

Performance of secure programs compiled with PICCO has been shown to be fast as illustrated in [13]. Our consequent work that treated the use of pointers to private data [11] also provides extensive analysis of data structures built via traditional pointer-based implementations. In addition, the compiler has already been used to build implementations that securely compute with private data in a number of applications such as processing of genomic data in [12, 8] and evaluation of statistical tests, which will be available in a forthcoming technical report.

References

[1] IT cloud services user survey, pt. 2: Top benefits & challenges. http://blogs.idc.com/ie/?p=210.

[2] M. Aliasgari and M. Blanton. Secure computation of hidden Markov models. In *International Conference on Security and Cryptography (SECRYPT)*, pages 242–253, July 2013.

[3] M. Aliasgari, M. Blanton, and F. Bayatbabolghani. Secure computation of hidden Markov models and secure floating-point arithmetic in the malicious model. *International Journal of Information Security*, to appear.

[4] M. Aliasgari, M. Blanton, Y. Zhang, and A. Steele. Secure computation on floating point numbers. In *Network and Distributed System Security Symposium (NDSS)*, 2013.

[5] M. Blanton and E. Aguiar. Private and oblivious set and multiset operations. *International Journal of Information Security*, 15(5):493–518, October 2016.

[6] M. Blanton and S. Saraph. Oblivious maximum bipartite matching size algorithm with applications to secure fingerprint identification. In *European Symposium on Research in Computer Security (ESORICS)*, pages 384–406, September 2015.

[7] M. Blanton, A. Steele, and M. Aliasgari. Data-oblivious graph algorithms for secure computation and outsourcing. In *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, pages 207–218, May 2013.

[8] A. Shahbazi, F. Bayatbabolghani, and M. Blanton. Private computation with genomic data for genome-wide association and linkage studies. In *International Workshop on Genomic Privacy and Security (GenoPri)*, November 2016.

[9] A. Yao. How to generate and exchange secrets. In *IEEE Symposium on Foundations of Computer Science*, pages 162–167, 1986.

[10] Y. Zhang and M. Blanton. Efficient secure and verifiable outsourcing of matrix multiplications. In *Information Security Conference (ISC)*, pages 158–178, October 2014.

[11] Y. Zhang, M. Blanton, and G. Almashaqbeh. Implementing support for pointers to private data in a general-purpose secure multi-party compiler. arXiv Report 1509.01763, 2015.

[12] Y. Zhang, M. Blanton, and G. Almashaqbeh. Secure distributed genome analysis for GWAS and sequence comparison computation. *BMC Medical Informatics and Decision Making Journal*, 15(5), December 2015.

[13] Y. Zhang, A. Steele, and M. Blanton. PICCO: A general-purpose compiler for private distributed computation. In *ACM Conference on Computer and Communications Security (CCS)*, pages 813–826, November 2013.

AFOSR Deliverables Submission Survey

Response ID:7289 Data

1.

Report Type

Final Report

Primary Contact Email

Contact email if there is a problem with the report.

mblanton@buffalo.edu

Primary Contact Phone Number

Contact phone number if there is a problem with the report

9372388366

Organization / Institution name

University at Buffalo, SUNY

Grant/Contract Title

The full title of the funded effort.

A Comprehensive Toolset for General-Purpose Private Computing and Outsourcing

Grant/Contract Number

AFOSR assigned control number. It must begin with "FA9550" or "F49620" or "FA2386".

FA9550-13-1-0066

Principal Investigator Name

The full name of the principal investigator on the grant or contract.

Marina Blanton

Program Officer

The AFOSR Program Officer currently assigned to the award

Tristan Nguyen

Reporting Period Start Date

03/01/2013

Reporting Period End Date

08/31/2016

Abstract

The over-reaching goal of this project is to provide the necessary tools and techniques for supporting general-purpose secure computation and outsourcing. The three main thrusts of the project are: (i) development of efficient techniques for securely working with standard data types, (ii) designing efficient data-oblivious algorithms and data structures suitable for secure computation and outsourcing, and (iii) building a compiler for translating a program written in a conventional programming language which is intended to handle private data into the corresponding secure distributed implementation that provably protects private data throughout program execution. This report summarizes the research findings of the project and scientific advances made towards each of the research thrusts throughout the project duration.

Distribution Statement

This is block 12 on the SF298 form.

Distribution A - Approved for Public Release

Explanation for Distribution Statement

If this is not approved for public release, please provide a short explanation. E.g., contains proprietary information.

SF298 Form

Please attach your SF298 form. A blank SF298 can be found here. Please do not password protect or secure the PDF The maximum file size for an SF298 is 50MB.

sf0298-completed.pdf

Upload the Report Document. File must be a PDF. Please do not password protect or secure the PDF . The maximum file size for the Report Document is 50MB.

final-report.pdf

Upload a Report Document, if any. The maximum file size for the Report Document is 50MB.

Archival Publications (published) during reporting period:

M. Aliasgari, M. Blanton, and F. Bayatbabolghani, Secure Computation of Hidden Markov Models and Secure Floating-Point Arithmetic in the Malicious Model, International Journal of Information Security, to appear.

M. Blanton and E. Aguiar, Private and Oblivious Set and Multiset Operations, International Journal of Information Security, Vol. 15, No. 5, pp. 493-518, Oct. 2016.

Y. Zhang, M. Blanton, and G. Almashaqbeh, Implementing Support for Pointers to Private Data in a General-Purpose Secure Multi-Party Compiler, arXiv Report 1509.01763, 2015.

M. Blanton and S. Saraph, Secure and Oblivious Maximum Bipartite Matching Size Algorithm with Applications to Secure Fingerprint Identification, European Symposium on Research in Computer Security (ESORICS), Sep. 2015.

New discoveries, inventions, or patent disclosures:
Do you have any discoveries, inventions, or patent disclosures to report for this period?

No

Please describe and include any notable dates

Do you plan to pursue a claim for personal or organizational intellectual property?

Changes in research objectives (if any):

Change in AFOSR Program Officer, if any:

Extensions granted or milestones slipped, if any:

AFOSR LRIR Number

LRIR Title

Reporting Period

Laboratory Task Manager

Program Officer

Research Objectives

Technical Summary

Funding Summary by Cost Category (by FY, $K)

	Starting FY	FY+1	FY+2
Salary			
Equipment/Facilities			
Supplies			
Total			

Report Document

Report Document - Text Analysis

Report Document - Text Analysis

Appendix Documents

2. Thank You

E-mail user

Nov 29, 2016 00:41:10 Success: Email Sent to: mblanton@buffalo.edu